Preview of Award 0903447 - Final Project Report

Cover

Federal Agency and Organization Element to Which Report is Submitted:	4900
Federal Grant or Other Identifying Number Assigned by Agency:	0903447
Project Title:	Qameleon: Hardware/software Co-operative Automated Tuning for Heterogeneous Architectures
PD/PI Name:	Hyesoon Kim, Principal Investigator Richard W Vuduc, Co-Principal Investigator
Submitting Official (if other than PD\PI):	N/A
Submission Date:	N/A
Recipient Organization:	Georgia Tech Research Corporation
Project/Grant Period:	08/01/2009 - 07/31/2013
Reporting Period:	08/01/2012 - 07/31/2013
Signature of Submitting Official (signature shall be submitted in accordance with agency specific instructions)	N/A

Accomplishments

*** What are the major goals of the project?**

The main goal of this project is to develop a framework that simplifies programming for heterogeneous platforms. The framework consists of (i) a run-time system to generate code that partitions and schedules work among heterogeneous processors, (ii) a general automated tuning mechanism based on machine learning and (iii) performance and power modeling techniques and profiling techniques to aid code generation.

*** What was accomplished under these goals (you must provide information for at least one of the 4 categories below)?**

Major Activities: The major outcomes from this project are

(i) Dynamic code compilation system for heterogeneous architecture[MICRO09]: We extended Qilin, which dynamically partitions data between CPU and GPU to support newer GPUs and also wide range of applications.

(ii) Automated code generation to partition and schedule work in heterogeneous architectures: [PPoPP10, PPoPP12] We studied several important kernels for autocode generation.

(iii) Automated architecture configuration tuning: [Report12, MICRO10b] We studied voltage and frequency should be configured in Report12. In MICRO10, as an example of configurable architecture, we proposed feedback-driven prefetching for GPUs.

(iv) Statisticalmodel-driven autotuning: [ISCA09, ISCA10, IPDPS10, PPoPP10, PPoPP12] we proposed various models for GPUs for autotuning.

(v) New profiling metrics and methods: We proposed several performance/power profiling and modeling work. [IPDPS13, PPOPP12, MICRO10a]

Specific Objectives: **We summarize our findings follow the key five categories.**

Significant Results:

1. **Dynamic code compilation system for heterogeneous architecture:**

In the first year, we have developed the basic dynamic code compilation system, Qilin. The work was presented in MICRO2009. The Qilin system distributes work between CPUs and GPUs based on a linear performance profiling function. In the second year, we extended Qilin to support newest GPUs, the Fermi, which have caches and more complex applications. That requires following modifications. (a) The basic Qilin splits array with only rows. We found that many scientific computing applications are written in column major so we have to support a column split feature. (b) To support complex data structures, Qilin first needs to copy all the data between CPUs and GPUs. This could increase the overhead of communications. In the third year, we also extended Qilin to consider architecture configurations and power and temperature.

2. **Automated code generation to partition and schedule the work in heterogeneous architecture**

We performed performance analysis by varying the input sizes. SpMV, GemV [PPoPP10], FMM [PPoPP12] benchmarks are evaluated. How to partition the work between CPUs and GPUs, how to reorganize the data structures, performance effects with various compiler optimizations are studied. Especially SpMV and GemV are studied using automated code generation frame. Different compiler optimizations, different blocking effects are also studied in the context of the auto code generation frame.

We also found out that a simple linear performance model works well for simple applications even for cache enabled GPUs. However the slopes and the constant offsets vary application by application. We also observed that we need at least 10% of sample data to train a correct performance model.

Most evaluated kernels are still simply following the SPMD programming style so there is no need for special scheduling policies for GPUs. However, for CPUs, because of NUMA characteristics, thread scheduling should be done more carefully.

3. **Automated architecture configuration tuning**

In the first year, we built the basic performance analytical model. We develop a modeling technique to find out when memory bandwidth can be saturated. In the second year, we have also implemented dynamic number of thread control for GPUs. We found that without power gating, the power savings that we can achieve are in 10% ranges but with power gating, we can achieve almost 30% energy savings for bandwidth saturated applications. We describe these results in ISCA10 paper.

We found that CPU power consumption can be reduced by DVFS but no more than 40%. This is because CPU power consumption is not only from cores, but also from memories and GPUs etc. Furthermore, even though cores can be power gated, caches, interconnection network and other supplementary logic should be on during the sleeping mode. Nonetheless, there are some points where DVFS can be beneficial.

We found that hardware prefetching can be still useful even for GPGPU applications. Hardware prefetching can be effective when there is not enough number of threads. Feedback driven hardware prefeching can improve performance of GPGPU applications by detecting when to insert prefetch or not at run-time[MICRO10b].

In the third year, we performed very intensive CPU and GPU architecture performance and power characterizations[REPORT12]. We use the Qilin benchmarks to support automatic partitioning of a computation among available CPU and GPU processors. The results suggest that how a combination of regression modeling of time and energy and Gaussian mixture models of power can serve as the foundation for model-guided autotuning with respect to all of time, energy, and power. The results show that if we only consider energy optimizations, optimizing for time is still the best metric, but when there is a power cap, time cannot be the only metric to optimize.

4. **Statistical model-driven autotuning**

We have shown that that highly-tuned implementation for a system based on conventional multicore processors can match or exceed the power/performance of the GPU system. The main significance of this result for the project is to establish what code transformations need to be included in the final system to be portable across both CPU and GPU architectures. We describe this result in [IPDPS 2010]. The secondary implication---that an *appropriate* (not naive) CPU implementation can match a GPU---will be of general interest to practitioners trying to decide whether to invest development resources into GPUs [HotPar 2010]. The PPoPP10 paper presented the autotuning frame model for SpMV on GPUs. Our results show

that having small sub-blocks inside a block can save data communication overhead, thereby increasing the performance.

5. **New profiling metrics and methods**

51.Performance models:

We improved GPU analytical models to read data from hardware performance counters and static code analysis in ISCA09. We implemented GPU analytical models for the Fermi architecture, which has caches. We found out that performance is strongly dependent on the number of instructions inside a kernel. Hence, it is critical to develop a mechanism to predict the number of instructions to support auto-tuning. We also discovered that monitoring temperatures can be a good metric to control power not only power itself. This is because much of power consumption is due to static power (leakage power) consumption.

We have developed dynamic memory profiling techniques that can help us to identify how to parallelize applications. Especially, we have focused on reducing memory profiling overhead. The results show that stride compression is a good way to reduce dynamic memory address profiling. [MICRO10a]

5.2 Power models

We recently began development of energy- and power-extensions to the "roofline model" of Williams, Waterman, and Patterson (in Communications of the ACM, 2009). The classical roofline model is an analytical tool designed to help programmers understand the performance potential of a given algorithm or code on a given architecture. Performance here means speed (inverse execution time). The critical need in current environments is to reduce energy and power, which directly motivates our extensions. Our "energy rooflines" aim to help programmers identify the upper-bounds in energy-efficiency (e.g., operations per Joule) possible for a given code.

Our initial energy roofline study has two components. The first derives an analytical model of the relationships among time, energy, and power. The inputs to the model are the computational intensity of the computation (ratio of useful work to memory transfers) and the time and energy costs per useful operation and per byte of communication. The output is an estimate of the upper-bound on energy-efficiency, to accompany the time-based roofline's upper-bound on time-efficiency (e.g., performance in FLOP/s). Using the PowerMon 2 system donated by Rob Fowler and Dan Bedard at UNC Chapel Hill / RENCI, we validated the basic form of this model experimentally for a variety of desktop, server, and consumer-grade systems based on Intel x86 processors, NVIDIA GPU processors, and ARM-based processors.

[IPDPS2013]

Key outcomes or Other achievements: Both PIs held tutorials on major compute architecture and supercomputing community conferences.

(i) 2011 HPCA tutorial, “Performance Analysis and Tuning for GPUs,”

(ii) 2010 Micro-42 tutorial, “Performance Analysis and Tuning for GPUs,”

(iii) 2010 TeraGrid’10 tutorial, “Performance Analysis and Tuning for GPUs,”

*** What opportunities for training and professional development has the project provided?**

Both PIs held tutorials on major compute architecture and supercomputing community conferences.

(i) 2011 HPCA tutorial, “Performance Analysis and Tuning for GPUs,”

(ii) 2010 Micro-42 tutorial, “Performance Analysis and Tuning for GPUs,”

(iii) 2010 TeraGrid’10 tutorial, “Performance Analysis and Tuning for GPUs,”

* Both PIs participate in a new Georgia Tech program that specifically targets underrepresented minority groups

* PI Kim taught in Korea during the summer in 2010

* PI Kim served as a mentor of DMP program sponsored by CRA-W in 2009

* PI Vuduc served as a science fair judge for the 2009 and 2010 Georgia Science and Engineering Fairs, in the mathematics category

*** How have the results been disseminated to communities of interest?**

PIs held tutorials (HPCA 2011, MICRO 2010) and also co-authored a book.

Products

Journals

Minjang Kim, Nagesh B. Lakshminarayana, Hyesoon Kim, Chi-Keung Luk (2012). SD3: An Efficient Dynamic Data-Dependence Profiling Mechanism. *IEEE Transactions on Computers*. (99), .

Status = AWAITING_PUBLICATION; Acknowledgment of Federal Support = No ; Peer Reviewed = Yes ; DOI: 10.1109/TC.2012.182

Books

Hyesoon Kim, Richard Vuduc, Sara Baghsorkhi, Jee Choi, and Wen mei Hwu. (2012). *Performance analysis and tuning for general purpose graphics processing units (GPGPU)* Mark Hill. Morgan & Claypool Publishers,.

Status = PUBLISHED; Acknowledgment of Federal Support = Yes ; Peer Reviewed = Yes ; DOI: http://www.morganclaypool.com/doi/abs/10.2200/S00451ED1V01Y201209CAC020

Book Chapters

Thesis/Dissertations

Conference Papers and Presentations

J. Choi, D. Bedard, R. Fowler, R. Vuduc (2013). *A roofline model of energy*. IEEE Parallel and Distributed Processing Symposium (IPDPS). Boston, USA.

Status = PUBLISHED; Acknowledgement of Federal Support = Yes

Jae Woong Sim, Aniruddha Dasgupta, Hyesoon Kim, and Richard Vuduc (2012). *A Performance Analysis Framework for Identifying Performance Benefits in GPGPU Applications*. 17th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP). New Orleans, LA.

Status = PUBLISHED; Acknowledgement of Federal Support = Yes

Jaekyu Lee, Nagesh B Lakshminarayana, Hyesoon Kim, Richard Vuduc (2010). *Hardware and Software Prefetching Mechanisms for GPGPU*. MICRO-43. Atlanta, GA.

Status = PUBLISHED; Acknowledgement of Federal Support = Yes

Minjang Kim, Hyesoon Kim, Chi-Keung Luk (2010). *SD3: A scalable Approach to Data-Dependence Profiling*. MICRO-43. Atlanta, GA.

Status = PUBLISHED; Acknowledgement of Federal Support = Yes

Hyesoon Kim, Anirudh Dasgupta, Richard Vuduc (2010). *Analytical Performance Models to Improve the Efficiency of GPU Computing*. GTC 2010. San Jose, CA.

Status = PUBLISHED; Acknowledgement of Federal Support = No

Richard Vuduc, Aparna Chandramowlishwaran, Jee Whan Choi, Murat Efe Guney, and Aashay Shringarpure (2010). *On the limits of GPU acceleration*. USENIX Wkshp. Hot Topics in Parallelism (HotPar),. Berkeley, CA.

Status = PUBLISHED; Acknowledgement of Federal Support = Yes

Aparna Chandramowlishwaran, Samuel Williams, Leonid Oliker, Ilya Lashuk, George Biros, and Richard Vuduc (2010). *Optimizing and tuning the fast multipole method for state-of-the-art multicore architectures*. IEEE Int'l. Parallel and Distributed Processing Symp. (IPDPS). Atlanta, GA.

Status = PUBLISHED; Acknowledgement of Federal Support = Yes

Jee Whan Choi, Amik Singh, and Richard W. Vuduc (2010). *Model-driven autotuning of sparse matrix-vector multiply on GPUs*. . ACM SIGPLAN Symp. Principles and Practice of Parallel Programming (PPoPP),. Bangalore, India,.

Status = PUBLISHED; Acknowledgement of Federal Support = Yes

Sunpyo Hong, Hyesoon Kim (2010). *An Integrated GPU Power and Performance Model*. ISCA-37. France.

Status = PUBLISHED; Acknowledgement of Federal Support = Yes

Chi-Keung Luk, Sunpyo Hong, Hyesoon Kim (2009). *Qilin: Exploiting Parallelism on Heterogeneous Multiprocessors with Adaptive Mapping*. MICRO 2009. New York, NY.

Status = PUBLISHED; Acknowledgement of Federal Support = Yes

Other Publications

Gurbinder Singh, Puyan Lotfi, Kent Czechowski, Hyesoon Kim, Richard Vuduc (2012). *Modeling Time, Power, and Energy of Tunable Programs on Heterogeneous Platforms*. Technical report.

Status = PUBLISHED; Acknowledgement of Federal Support = Yes

Technologies or Techniques

- We developed profiling techniques that are integrated in dynamic code compilation systems
-
- We developed performance models that are used in automated code generation.

- We developed techniques to automatically partition the work between CPUs and GPUs and schedule them.
- We developed new profiling techniques that can predict GPU power behavior.
- We developed automated scheduling techniques that can distribute work between CPUs and GPUs using performance models.

Patents
Nothing to report.

Inventions
Nothing to report.

Licenses
Nothing to report.

Websites
Nothing to report.

Other Products
Nothing to report.

Participants

Research Experience for Undergraduates (REU) funding

What individuals have worked on the project?

Name	Most Senior Project Role	Nearest Person Month Worked
Sunpyo Hong	Graduate Student (research assistant)	4
Richard W Vuduc	Co PD/PI	1
Hyesoon Kim	PD/PI	1

What other organizations have been involved as partners?
Nothing to report.

Have other collaborators or contacts been involved? Y

Impacts

What is the impact on the development of the principal discipline(s) of the project?

Collectively, our project has developed new models that we believe can be used to design autotuning frameworks that target systems with GPGPU components. In particular, consider these examples:

- Paper [PPoPP10] is the first example of autotuning sparse matrix computations for GPGPUs.

- Paper [PPoPP12] is the first example of how to use modeling to quantify potential performance benefits. The prior state-of-the-art approaches focused instead on identifying hotspots. Knowledge of benefits could be used to steer an autotuning system toward regions of interest in the space of candidate program transformations.

Paper [ISCA10] is the first GPGPU power modeling. The paper provides details of power measurement data and a simple model to predict power behavior.

What is the impact on other disciplines?

- Many scientific applications start to utilize GPUs to calculate large data applications such as DNA analysis, weather predictions etc. The developed techniques can improve the computation speed to allow fast computations or increase the accuracy of predictions. More specifically, performance models can be used in optimizing other scientific applications such as computational biology or molecular dynamics. For example, PPoPP11 paper's model can be extended to provide optimization suggestions.

What is the impact on the development of human resources?

Students working on this project received exposure to multiple subdisciplines, namely, computer architecture, compilers, and parallel algorithms. Papers [IPDPS13], [SRC-Report12], [PPOPP12], [MICRO10b], [GTC10], [IPDPS10] in particular involved authors from all three sub disciplines.

In addition, this project helped expose two undergraduates to research: Amik Singh, co-author on [PPoPP10], now a graduate student at UC Berkeley in Computer Science; and Gurbinder Singh [SRC-Report12], now a graduate student at UT Austin in Computer Science.

What is the impact on physical resources that form infrastructure?

- Report [SRC-Report12] shows that statistical modeling approaches, previously used to autotune with respect to execution time, can also be used to autotune with respect to energy and caps on power.

What is the impact on institutional resources that form infrastructure?
Nothing to report.

What is the impact on information resources that form infrastructure?
Nothing to report.

What is the impact on technology transfer?

Some part of the research was done with industry folks especially CK Luk at Intel.

Students spent one or two semesters at Intel to develop performance and power models for other architectures based on the developed techniques.

PIs have interacted heavily with other industry partners such as NVIDIA, Qualcomm, and AMD and have given talks at these companies.

What is the impact on society beyond science and technology?

Applications that use large data or large computation can get benefits from running on CPUs and GPUs using the proposed techniques. Many of these applications are often very closely related to various aspects of human life. For example, weather prediction applications can increase the accuracy of weather predictions that will improve the quality of life.

Changes

Changes in approach and reason for change
Nothing to report.

Actual or Anticipated problems or delays and actions or plans to resolve them

Nothing to report.

Changes that have a significant impact on expenditures
Nothing to report.

Significant changes in use or care of human subjects
Nothing to report.

Significant changes in use or care of vertebrate animals
Nothing to report.

Significant changes in use or care of biohazards
Nothing to report.

Special Requirements

Responses to any special reporting requirements specified in the award terms and conditions, as well as any award specific reporting requirements.
Nothing to report.